UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/986,019	11/07/2001	Stefan Dyckerhoff	0023-0051	4743

44987 7590 08/24/2004

HARRITY & SNYDER, LLP
11240 WAPLES MILL ROAD
SUITE 300
FAIRFAX, VA 22030

EXAMINER	
GERSTL, SHANE F	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 08/24/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	09/986,019	DYCKERHOFF ET AL.
	Examiner	Art Unit
	Shane F Gerstl	2183

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *07 November 2001*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-30* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-30* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *07 November 2001* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-30 have been examined.

Papers Received

2. Receipt is acknowledged of application papers submitted, where the papers have been placed of record in the file.

Drawings

3. The drawings are objected to as failing to comply with 37 CFR 1.84(p)(5) because they include the following reference character(s) not mentioned in the description: 607. Corrected drawing sheets in compliance with 37 CFR 1.121(d), or amendment to the specification to add the reference character(s) in the description in compliance with 37 CFR 1.121(b) are required in reply to the Office action to avoid abandonment of the application. Any amended replacement drawing sheet should include all of the figures appearing on the immediate prior version of the sheet, even if only one figure is being amended. The replacement sheet(s) should be labeled "Replacement Sheet" in the page header (as per 37 CFR 1.84(c)) so as not to obstruct any portion of the drawing figures. If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

Claim Objections

4. Claim 1 is objected to because of the following informalities: lines 4-5 of the claim read, "a program counter configured to read instructions from the plurality of instruction memories," when it is clear from the specification and figure 4 that the

program counter itself is not configured to read instructions, but instead the program counter is configured to provide an address to the plurality of memories to read from.

5. Claim 16 is objected to because of the following informalities: The article "the" should be added so the claim reads, "...connected to the evaluation component."

6. Appropriate correction is required.

7. Claim 2 is objected to under 37 CFR 1.75(c), as being of improper dependent form for failing to further limit the subject matter of a previous claim. Applicant is required to cancel the claim(s), or amend the claim(s) to place the claim(s) in proper dependent form, or rewrite the claim(s) in independent form. As shown in MPEP 2106 Section II (C), language that suggests steps but does not limit a claim to a particular structure does not limit the scope of a claim. This language includes statements of intended use. The language of claim 2 simply states that execution is related to packet header information but gives no structural information on this relation. The statement merely gives suggestion that operation of the execution unit somehow relates or is to be used with packet header processing. Therefore, the examiner is not giving claim 2 patentable weight.

Claim Rejections - 35 USC § 112

8. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

9. Claim 25 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

10. Claim 25 recites the limitation "the operation" in line 1 of the claim. There is insufficient antecedent basis for this limitation in the claim. The parent claim has multiple operations but not a single operation. The examiner is taking the claim to read "...wherein the operations include..."

Claim Rejections - 35 USC § 102

11. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

12. Claims 21-22, 27, and 29-30 are rejected under 35 U.S.C. 102(e) as being anticipated by Nemirovsky (6,477,562).

13. In regard to claim 21, Nemirovsky discloses a method for processing a packet (column 5, lines 10-21 and column 12, lines 27-31) to determine control information for the packet, the method comprising:

a. reading a plurality of instructions; [Column 7, lines 3-7 show that instructions are fetched from the instruction cache (memory)]

b. generating a predicted address based on a predetermined one of the plurality of instructions; [Column 6, lines 9-16 show that branch prediction is used and thus a predicted address is generated and is inherently based on a branch instruction.]

c. evaluating the read instructions; [Column 5, lines 10-21 show that the priority (evaluation results) is dynamically determined or evaluated and is based on each stream and thus the instructions therein.]

d. selecting one of the read instructions based on the evaluations; [Column 7, lines 57-60 show that since there are multiple streams, there is a mechanism for selecting the stream, which is then executed by an execution unit of figure 2 (elements 207-210).]

e. and performing operations related to determining the control information for the packet based on the selected instruction, the operations including generating a true next address for reading instructions. [Branch prediction inherently must be checked with the outcome of the execution of the branch so it is known whether or not the prediction was correct or accurate so that the correct code is certain to be executed. Since the system has a branch execution unit (figure 2, element 207), this is where the branch execution takes place that reveals the true next instruction address or program counter address.]

14. In regard to claim 21, Nemirovsky discloses the method of claim 21, wherein evaluating the read instructions is performed based on a field in the instructions that specifies a logical operation that is to be performed. [The enclosed definition of "header" shows that it is a portion of the packet that contains source and destination addresses and other fields. The included IEEE definition shows that a header is all of this plus also gives the type of frame or packet. Column 12, lines 19-34 show that priority is set or

evaluated based on the type of thread or packet being handled. Since the circuitry to do so is inherently made up of logic gates, a logic function is performed.]

15. In regard to claim 27, Nemirovsky in view of Hennessy discloses a pipelined processing device comprising:

a. means for reading a plurality of packet processing instructions from instruction memory; [Column 7, lines 3-7 show that instructions are fetched or read from the instruction cache (memory).]

b. and means for selecting one of the read instructions for execution based on a priority encoding of evaluation results related to each of the read instructions. [Column 7, lines 57-60 show that since there are multiple streams, there is a mechanism for selecting the stream and this is based on priority. Column 5, lines 10-21 show that the priority (evaluation results) is dynamically determined or evaluated and is based on each stream and thus the instructions therein.]

16. In regard to claim 29, Nemirovsky discloses the pipelined processing device of claim 27, further comprising: means for storing a value that designates an address to the instruction memory. [Column 7, lines 3-7 show that instructions are fetched from the instruction cache (memory). Column 7, lines 14-19 show that a program counter exists for multiple streams or threads. The included dictionary definition of "program counter" shows that a program counter points to the next instruction to be fetched or read.]

17. In regard to claim 30, Nemirovsky discloses the pipelined processing device of claim 29, further comprising: means for generating a predicted next value to store in the

means for storing; and means for generating a true next value to store in the means for storing.

Claim Rejections - 35 USC § 103

18. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

19. Claims 1-20, 23-26, and 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over Nemirovsky in view of Hennessy (Computer Organization and Design).

20. In regard to claim 1,

a. Nemirovsky discloses a pipelined processor comprising:

i. An instruction memory (figure 2, element 202);

ii. a program counter configured to provide the address for reading instructions from the plurality of instruction memories; [Column 7, lines 3-7 show that instructions are fetched from the instruction cache (memory). Column 7, lines 14-19 show that a program counter exists for multiple streams or threads. The included dictionary definition of "program counter" shows that a program counter points to the next instruction to be fetched or read.]

iii. a priority encoder configured to select one of the instructions based on evaluation results generated from the instructions read from the

instruction memories; [Column 7, lines 57-60 show that since there are multiple streams, there is a mechanism for selecting the stream and this is based on priority. Column 5, lines 10-21 show that the priority (evaluation results) is dynamically determined or evaluated and is based on each stream and thus the instructions therein. The included definitions of "encoder" and "encode" show that an encoder is simply a circuit that converts data into a given format and the selecting mechanism above does so by converting multiple inputs into a single selected output.]

iv. and an execution unit configured to receive the selected one of the instructions and to perform operations indicated by the selected instruction (figure 2, elements 207-210).

b. Nemirovsky does not explicitly disclose

v. a plurality of memories;

vi. pipelining in general nor a first pipeline stage including the memories and program counter and a second pipeline stage including the priority encoder and execution unit.

c. Hennessy has taught in figure 6.12 on page 452 a first pipeline stage (IF) including an instruction memory and program counter (PC) and a second pipeline stage (EX) that contains an execution unit (ALU) and logic to select appropriate instruction data (MUX).

d. Hennessy teaches on page 436 that pipelining speeds up execution. This execution speed-up taught by Hennessy would have motivated one of ordinary

skill in the art to modify the design of Nemirovsky to use the pipeline disclosed in Hennessy. With such a pipeline in Nemirovsky, the instruction cache (element 202 of figure 2) and the fetch unit (within element 203) would be in a first pipeline stage while the execution units (elements 207-210), issue network (element 6), and instruction scheduler (element 5) would be in a second pipeline stage. Nemirovsky discloses an instruction cache memory in figure 2, element 202. While a plurality of instruction memories is not explicitly taught, fetching multiple threads and streams from the instruction cache memory is taught (column 6, line 65 – column 7, line 3 for example). The inclusion of a plurality of instruction cache memories to perform the same function as a single instruction cache memory provides no new or unexpected result over the prior art. Therefore, one of ordinary skill in the art would have found it obvious to duplicate the instruction cache memory, creating a plurality of instruction cache memories to fetch from (see In re Harza, 274 F.2d 669, 671, 124 USPQ 378, 380 (CCPA 1960)).

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Nemirovsky to use the pipeline of Hennessy so the processor is sped-up as taught by Hennessy and to use multiple instruction cache memories since no operational change is achieved by duplicating the instruction cache memory.

21. In regard to claim 2, Nemirovsky in view of Hennessy discloses the pipelined processor of claim 1, wherein the operation performed by the execution unit relates to processing of packet header information. [Since the claim has not been given patentable weight, it is unpatentable for the same reasons as parent claim 1.]

22. In regard to claim 3, Nemirovsky in view of Hennessy discloses the pipelined processor of claim 1, further comprising: a packet header buffer connected to the priority encoder and the execution unit. [The enclosed definition of "header" shows that it is a portion of the packet that contains source and destination addresses and other fields. Therefore, the header, being a part of the packet, must be stored like the packet in the memories (instruction cache), pipeline buffers (figure 6.12 of Hennessy), and prefetch buffers of figure 2.]

23. In regard to claim 4, Nemirovsky in view of Hennessy discloses the pipelined processor of claim 1, wherein the instructions are read from a memory address equal to the value of the program counter (as shown above).

24. In regard to claim 5, Nemirovsky in view of Hennessy discloses the pipelined processing processor of claim 1, wherein the second pipeline stage further comprises: an evaluation component corresponding to each of the instruction memories, the evaluation components generating the evaluation results based on each of the instructions read from the instruction memories. [Column 5, lines 10-21 show that the priority codes (evaluation results as above) are associated with the streams and dynamically determined by a priority controller or evaluation component. Since there is only one priority controller in the embodiment of figure 2, element 9, it must be associated with each of the plurality of instruction memories.]

25. In regard to claim 6, Nemirovsky in view of Hennessy discloses the pipelined processor of claim 5, wherein the evaluation components generate the evaluation results based on a logical operation dictated by the instructions. [Column 6, lines 50-60

show that the priority control unit changes or evaluates the priority and is comprised of logic and thus performs a logical operation. As shown above, the priority is associated with the instructions in the streams and in a specific embodiment of column 12, lines 27-31, for example, it is shown that the type of thread or stream (and thus the type of instructions) control or dictate the priority.]

26. In regard to claim 7, Nemirovsky in view of Hennessy discloses the pipelined processor of claim 1 further comprising:

a. a branch prediction component (column 6, lines 9-16) configured to generate a predicted program counter value based on instructions read from one of the instruction memories; [Since branch prediction gives a prediction of a branch instruction, the prediction is inherently based on a branch instruction, which must be read from an instruction memory. Also, since the system of Nemirovsky in view of Hennessy points to instructions to fetch with a program counter and branch prediction inherently produces a predicted next address the (whether the branch is taken or not taken), the prediction must produce a predicted program counter.]

b. wherein the execution unit generates a true program counter value based on the selected instruction and generates an indication of whether the predicted program counter value is accurate. [Branch prediction inherently must be checked with the outcome of the execution of the branch so it is known whether or not the prediction was correct or accurate so that the correct code is certain to be executed. Since the system has a branch execution unit (figure 2, element

207), this is where the branch execution takes place that reveals the true next instruction address or program counter address.]

27. In regard to claim 8, Nemirovsky in view of Hennessy the pipelined processor of claim 1, wherein the second pipeline stage further comprises: a multiplexer configured to receive the read instructions and to forward the selected one of the instructions to the execution unit based on a signal from the priority encoder. [As described above, the selection of the instructions in the second pipeline stage is based on a priority and the selection unit chooses an instruction at one input from many and outputs it (the definition of a multiplexer) for execution.]

28. In regard to claim 9, Nemirovsky in view of Hennessy the pipelined processor of claim 1, further comprising: a plurality of memory elements implemented as an interface between the first and second pipeline stages. [The pipeline of Hennessy in figure 6.12 shows that there are memory elements between each pipeline stage.]

29. In regard to claim 10,

a. Nemirovsky discloses a network device comprising:

i. a physical interface configured to receive packets from and transmit packets to a network; [Column 12, lines 20-24 show that flows of packets are received from and transmitted to a network.]

ii. and a processing unit configured to store the received packets (Figure 2) and to examine header information of the packets, [The enclosed definition of "header" shows that it is a portion of the packet that contains source and destination addresses and other fields. The included

IEEE definition shows that a header is all of this plus also gives the type of frame or packet. Column 12, lines 19-34 show that priority is set based on the type of thread or packet being handled. Since this type is stored in the header of the packet, the header must be examined.]

iii. the processing unit including a processing engine that comprises:

(1) reading a plurality of packet processing instructions from instruction memory, [Column 7, lines 3-7 show that instructions are fetched from the instruction cache (memory)]

(2) selecting one of the instructions for execution; [Column 7, lines 57-60 show that since there are multiple streams, there is a mechanism for selecting the stream, which is then executed by an execution unit of figure 2 (elements 207-210).

b. Nemirovsky does not explicitly disclose the processing engine being a pipelined packet processing engine that comprises:

iv. a first pipeline stage containing the read function;

v. and a second pipeline stage containing the select function.

c. Hennessy has taught in figure 6.12 on page 452 an instruction fetch (read) pipeline stage (IF) including an instruction memory and a second pipeline stage (EX) that contains an execution unit (ALU) and logic to select appropriate instruction data (MUX).

d. Hennessy teaches on page 436 that pipelining speeds up execution. This execution speed-up taught by Hennessy would have motivated one of ordinary

skill in the art to modify the design of Nemirovsky to use the pipeline disclosed in Hennessy. With such a pipeline in Nemirovsky, the instruction cache (element 202 of figure 2) and the fetch unit (within element 203) would be in a first pipeline stage while the execution units (elements 207-210), issue network (element 6), and instruction scheduler (element 5) would be in a second pipeline stage.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Nemirovsky to use the pipeline of Hennessy so the processor is sped-up as taught by Hennessy.

30. In regard to claim 11, Nemirovsky in view of Hennesy discloses the network device of claim 10, wherein the network device is a router (column 12, lines 19-31 and column 5, lines 10-21).

31. In regard to claim 12, Nemirovsky in view of Hennessy discloses the network device of claim 10, wherein the first pipeline stage comprises: a program counter configured to store a program address value used to read the packet processing instructions from the instruction memories. [Column 7, lines 3-7 show that instructions are fetched from the instruction cache (memory). Column 7, lines 14-19 show that a program counter exists for multiple streams or threads. The included dictionary definition of "program counter" shows that a program counter points to the next instruction to be fetched or read. The pipeline of Hennessy shows that the program counter (PC) is in the first stage.]

32. In regard to claim 13, Nemirovsky in view of Hennessy discloses the network device of claim 10, wherein the second pipeline stage comprises: a priority encoder

configured to select the one of the packet processing instructions based on evaluation results generated from the packet processing instructions read from the instruction memories, and an execution unit (figure 2, elements 207-210) configured to receive the selected one of the packet processing instructions and to perform operations indicated by the selected packet processing instruction. [Column 7, lines 57-60 show that since there are multiple streams, there is a mechanism for selecting the stream and this is based on priority. Column 5, lines 10-21 show that the priority (evaluation results) is dynamically determined or evaluated and is based on each stream and thus the instructions therein. The included definitions of "encoder" and "encode" show that an encoder is simply a circuit that converts data into a given format and the selecting mechanism above does so by converting multiple inputs into a single selected output.]

33. In regard to claim 14, Nemirovsky in view of Hennessy discloses the network device of claim 13, wherein the instructions read from the instruction memories are read from a memory address equal to the value of the program counter (as shown above).

34. In regard to claim 15, Nemirovsky in view of Hennessy discloses the network device of claim 13, wherein the second pipeline stage further comprises: an evaluation component corresponding to each of the instruction memories, the evaluation components generating the evaluation results based on each of the instructions read from the instruction memories. [Column 5, lines 10-21 show that the priority codes (evaluation results as above) are associated with the streams and dynamically determined by a priority controller or evaluation component. Since there is only one

priority controller in the embodiment of figure 2, element 9, it must be associated with each of the plurality of instruction memories.]

35. In regard to claim 16, Nemirovsky in view of Hennessy discloses the network device of claim 15, further comprising: a packet header buffer connected to the evaluation component. [The enclosed definition of "header" shows that it is a portion of the packet that contains source and destination addresses and other fields. Therefore, the header, being a part of the packet, must be stored like the packet in the memories (instruction cache), pipeline buffers (figure 6.12 of Hennessy), and prefetch buffers of figure 2.]

36. In regard to claim 17, Nemirovsky in view of Hennessy discloses the network device of claim 15, wherein the evaluation components generate the evaluation results based on a logical operation dictated by the packet processing instructions. [Column 6, lines 50-60 show that the priority control unit changes or evaluates the priority and is comprised of logic and thus performs a logical operation. As shown above, the priority is associated with the instructions in the streams and in a specific embodiment of column 12, lines 27-31, for example, it is shown that the type of thread or stream (and thus the type of instructions) control or dictate the priority.]

37. In regard to claim 18, Nemirovsky in view of Hennessy discloses the network device of claim 10, further comprising:

a. a branch prediction component (column 6, lines 9-16) configured to generate a predicted program counter value based on packet processing instructions read from one of the instruction memories, [Since branch prediction

gives a prediction of a branch instruction, the prediction is inherently based on a branch instruction, which must be read from an instruction memory. Also, since the system of Nemirovsky in view of Hennessy points to instructions to fetch with a program counter and branch prediction inherently produces a predicted next address the (whether the branch is taken or not taken), the prediction must produce a predicted program counter.]

b. wherein an execution unit generates a true program counter value based on the selected packet processing instruction and generates an indication of whether the predicted program counter value is accurate. [Branch prediction inherently must be checked with the outcome of the execution of the branch so it is known whether or not the prediction was correct or accurate so that the correct code is certain to be executed. Since the system has a branch execution unit (figure 2, element 207), this is where the branch execution takes place that reveals the true next instruction address or program counter address.]

38. In regard to claim 19, Nemirovsky in view of Hennessy discloses the network device of claim 10, wherein the second pipeline stage further comprises: a multiplexer configured to receive the read packet processing instructions and to forward the selected one of the packet processing instructions to an execution unit based on a signal from the priority encoder. [As described above, the selection of the instructions in the second pipeline stage is based on a priority and the selection unit chooses an instruction at one input from many and outputs it (the definition of a multiplexer) for execution.]

39. In regard to claim 20, Nemirovsky in view of Hennessy discloses the network device of claim 10, further comprising: a plurality of timing buffers implemented as an interface between the first and second pipeline stages. [The pipeline of Hennessy in figure 6.12 shows that there are memory elements between each pipeline stage.]

40. In regard to claim 23,

a. Nemirovsky discloses the method of claim 21,

b. Nemirovsky does not explicitly disclose wherein the method is performed in two pipelined stages.

c. Hennessy has taught in figure 6.12 on page 452 a pipelined processor with at least two pipelined stages.

d. Hennessy teaches on page 436 that pipelining speeds up execution. This execution speed-up taught by Hennessy would have motivated one of ordinary skill in the art to modify the design of Nemirovsky to use the pipeline disclosed in Hennessy.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Nemirovsky to use the pipeline of Hennessy so the processor is sped-up as taught by Hennessy.

41. In regard to claim 24, Nemirovsky in view of Hennessy discloses the method of claim 21, wherein the selecting of one of the read instructions is performed as a priority selection based on a read instruction that evaluates to a logic true value. [Column 7, lines 57-60 show that since there are multiple streams, there is a mechanism for selecting the stream and this is based on priority. Column 5, lines 10-21 show that the

priority (evaluation results) is dynamically determined or evaluated and is based on each stream and thus the instructions therein. The instruction that is being selected is being interpreted as the logic true value since it has the highest evaluated priority.]

42. In regard to claim 25, Nemirovsky in view of Hennessy discloses the method of claim 21, wherein the operation includes an extract instruction that is used to extract designated information from the packet into a memory. [Figure 2 illustrates a load/store unit and thus load instructions are executed, which inherently extract information from memory based on information extracted from the load instruction of the packet and then stored in a register or other local memory.]

43. In regard to claim 26, Nemirovsky in view of Hennessy discloses the method of claim 21, wherein the operation includes a write instruction used to write information contained in a field of the write instruction to a memory. [Figure 2 illustrates a load/store unit and thus store instructions are executed, which inherently write information from a field indicated by the instruction into memory.]

44. In regard to claim 28, Nemirovsky discloses

a. The pipelined processing device of claim 27,

b. Nemirovsky does not disclose wherein the means for reading and the means for selecting are implemented as first and second stages of the pipeline, respectively.

c. Hennessy has taught in figure 6.12 on page 452 an instruction fetch (read) pipeline stage (IF) including an instruction memory and a second pipeline stage

(EX) that contains an execution unit (ALU) and logic to select appropriate instruction data (MUX).

d. Hennessy teaches on page 436 that pipelining speeds up execution. This execution speed-up taught by Hennessy would have motivated one of ordinary skill in the art to modify the design of Nemirovsky to use the pipeline disclosed in Hennessy. With such a pipeline in Nemirovsky, the instruction cache (element 202 of figure 2) and the fetch unit (within element 203) would be in a first pipeline stage while the execution units (elements 207-210), issue network (element 6), and instruction scheduler (element 5) would be in a second pipeline stage.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Nemirovsky to use the pipeline of Hennessy so the processor is sped-up as taught by Hennessy.

Conclusion

45. The following is text cited from 37 CFR 1.111(c): In amending in reply to a rejection of claims in an application or patent under reexamination, the applicant or patent owner must clearly point out the patentable novelty which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the objections made. The applicant or patent owner must also show how the amendments avoid such references or objections.

46. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. The following references have been cited to further show the art with respect to packet processing and priority scheduling in general.

US Pat No 5,627,982 to Hirata teaches multiple memory streams with instruction selection scheduling based on priority.

US Pat No 6,760,337 to Snyder, II shows communications packet scheduling based on priority.

US Pat No 6,160,809 to Adiletta discloses a packet processing system with packets stored in multiple memories wherein the headers of each packet are analyzed.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Shane F Gerstl whose telephone number is (703)305-7305. The examiner can normally be reached on M-F 6:45-4:15 (First Friday Off).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703)305-9712. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Shane F Gerstl
Examiner
Art Unit 2183

SFG
August 19, 2004

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100